SERIAL PERIPHERAL INTERFACE AND RELATED METHODS

Field of the Invention

[0001] The present invention relates in general to serial interfaces for data communications between a control unit and external peripherals, and, more particularly, to serial peripheral interfaces (SPIs).

Background of the Invention

[0002] Serial interfaces are widely used for data communications between a control unit and its peripherals. In particular, SPIs are commonly associated with microprocessors and their peripherals. A block diagram of a typical prior art SPI interface is shown in FIG. 1. A control circuit PIN CONTROL LOGIC has a pin SCK for either receiving or outputting a clock signal, depending on whether the SPI is in a slave or in a master device, respectively. It also includes a pair of input/output pins MISO (master in/slave out) and MOSI (master out/slave in), as well as another pin SS (Slave select) for allowing external configuration of the SPI interface as a slave device, or for selecting an external peripheral.

[0003] If the interface is in a master device, the pin MISO serves as an input for signals coming from the slave device, and the pins MOSI and SS serve as output pins providing signals for the slave device. The opposite happens when the SPI interface is in a slave device.

[0004] A pair of FIFO memory buffers TRANSMIT BUFFER and RECEIVE BUFFER respectively store data to be transmitted and received data. These buffers are connected to a bus DATA IN for data to be transmitted to the peripherals, an addresses bus ADDR for memory locations where data in transit is to be stored, and a bus DATA OUT for data received from the connected peripherals. Shift registers RECEIVE SHIFT REGISTER and TRANSMIT SHIFT REGISTER serially receive and transfer, respectively, to the circuit PIN CONTROL CIRCUIT received data or data to be transmitted.

[0005] The buffers are connected to circuit blocks (i.e., counters) POINTER and CURRENT POINTER which generate a pointer to the memory location to be read. More particularly, the counter POINTER increments the current value of the pointer stored in the register CURRENT POINTER. The comparator COMP compares the current value of the pointer with a predetermined maximum value, generated by the block END POINTER, and eventually indicates that all bits of the word have been read when the maximum value is reached.

[0006] The control register CTRL exchanges information for configuring the SPI interface with the control circuit PIN CONTROL LOGIC. It may also load a certain default value in the circuit CURRENT POINTER. Another kind of SPI interface is the so-called "queued SPI" (QSPI), disclosed in U.S. Patent Nos. 4,816,996 and 4,958,277 to S.C. Hill et al., which is schematically illustrated in FIG. 2. It differs from the interface of FIG. 1 in that the data to be transmitted and received is not stored in a pair of FIFO buffers. Rather, this data is stored in a RAM memory together with commands to be executed.

[0007] More particularly, the RAM memory is used to store data in transit between interfaces and peripherals, as well as control commands used to specify the length and destination of data together with other parameters associated with each transfer. The control circuit has chip select pins CS[0-n] for selecting a peripheral to communicate with when the interface is in a master device. The pins CS are not used when the interface is in a slave device. The information necessary for determining the desired chip select value on the relative pin is provided to the control circuit by the RAM memory.

[0008] The RAM memory is organized in three memory sections RECEIVE RAM, TRANSMIT RAM and COMMAND RAM, which are respectively for received data, data to be transmitted, and commands for configuring the interface in the desired mode. The pointer provided to the block CURRENT POINTER is used for pointing, at the same time, to a certain word in the memory section RECEIVE RAM, a certain word in the memory section TRANSMIT RAM, and a byte in the memory section COMMAND RAM identified by the address specified on the bus ADDR. The configuration commands to be executed are provided to the interface by an external controller each time that data passes through the interface.

[0009] This architecture is more advantageous than illustrated in FIG. 1 in that each time data to be transmitted and a configuration command have been written in the RAM memory, the interface may execute the programmed transfers simply by incrementing the pointer generated by the CURRENT POINTER without the intervention of an external controller. Moreover, when the data to be transmitted has been written in the memory, it may be retransmitted again many times to different peripherals without any further write operations in the memory simply by changing the bits of the configuration command that identify the different destinations.

[0010] However, QSPI interfaces require an external controller to write a command in the memory section COMMAND of the RAM memory each time data is exchanged, transmitted to, or received from a peripheral. This is disadvantageous because of the resulting burden placed on the external controller, making its architecture more complex.

Summary of the Invention

[0011] An object of the present invention is to provide a serial interface for communicating with peripherals that does not require an external controller to provide commands for each data exchange with the connected peripherals.

[0012] This and other objects, features, and advantages in accordance with the invention are provided by an interface which may include a RAM memory coupled to at least one external data bus for transferring data to be transmitted or received from a peripheral. The RAM memory may also be coupled to an external address bus over which addresses are communicated for storing transmitted or received data in the memory. More particularly, the RAM memory may include a memory section for storing data to be transmitted, a memory section for storing received data, and a memory section for storing configuration commands of the interface.

[0013] The interface may also include a circuit or means for generating pointers to addresses in the memory sections, and a circuit or means for serially transferring data from or to at least one peripheral connected to the interface. This circuit is coupled to the memory and executes the configuration command pointer in the memory section for storing commands. A relative control register may also be coupled to the memory and to the serial transfer circuit for controlling the transfer of data to be transmitted or received.

[0014] The interface of the present invention advantageously does not require that an external controller provide configuration commands for data to be transmitted or received. According to the invention, the memory sections for storing data are divided in distinct memory spaces, and each memory space stores data pertaining to a respective peripheral connected to the interface. The memory section which stores commands includes all of the configuration commands of the interface for communicating with peripherals connected thereto.

[0015] The interface may also include an additional circuit for generating addresses to the memory section storing the configuration commands. This circuit receives addresses provided on the external address bus and generates corresponding addresses based thereon at which the appropriate configuration commands to be executed are stored.

[0016] A method aspect of the invention is for managing a serial peripheral interface of the invention and may include initializing the interface by loading, in the memory section dedicated to storing commands, the configuration commands of the interface for communicating with the peripherals connected thereto. The method may further include associating with each connected peripheral respective memory spaces of the memory sections for storing data in transit to or from the peripheral. Moreover, for each address on the address bus, corresponding addresses of the memory section including the configuration commands may be generated.

[0017] For transmitting data to a certain peripheral, the method may further include sending an address on the address bus of the memory section for storing data associated with the peripheral to be transmitted, and configuring the interface according to the commands stored in the corresponding addresses of the memory section including the configuration commands. Further, for receiving data from a certain peripheral, the method may include sending an address on the address bus of the memory section for storing received data associated with the peripheral. Also, the interface may be configured according to the commands stored at the corresponding addresses of the memory section including the configuration commands.

[0018] In practice, data conveyed through the interface is written at different addresses depending on the peripheral with which it is associated. Therefore, it is sufficient to specify the memory address at which the data to be transmitted is stored, or at which received data is to be stored, for identifying the peripheral to which the data belongs. Thus, the appropriate interface configuration commands may be executed, because the configuration memory has already been provided with the configuration commands that may be used for communicating with the connected peripherals. In this way, the external controller no longer need be engaged in any interface configuration task during data exchanges.

Brief Description of the Drawings

[0019] The various aspects and advantages of the invention will become more evident through a detailed description with reference to the attached drawings, in which:

[0020] FIG. 1 is a schematic block diagram of the architecture of a prior art SPI interface;

[0021] FIG. 2 is a schematic block diagram of the architecture of a prior art queued SPI; and

[0022] FIG. 3 is a schematic block diagram of an interface in accordance with the present invention having a memory section including configuration commands of the interface for peripherals connected thereto.

Detailed Description of the Preferred Embodiments

[0023] The SPI interface of the present invention is illustrated in FIG. 3. As may be observed, its general architecture is similar to that of a Queued SPI. Yet, instead of the block COMMAND RAM, a configuration memory MODE RAM and a pointer generator POINTER RAM which provides pointers thereto. The size of the configuration memory is determined by the number of commands to be executed, and thus by the number of external peripherals that may be coupled through the interface. Each of the peripherals is selected by activating a respective chip select CS[0-n].

[0024] The interface of the present invention is managed using an initialization phase in which the memory MODE RAM is programmed with all the required configuration commands for the peripherals associated therewith. This is done because the peripherals connected to the device including the interface are recognized only after initialization of the interface. Thereafter, distinct memory spaces of the memory sections TRANSMIT RAM, RECEIVE RAM are dedicated for storing data to be transmitted and received, respectively, for each connected peripheral. Thus, based upon the address on the bus ADDR, it may be determined whether data to be transmitted or received needs to be stored, and which peripheral it is associated with.

[0025] For example, assume the interface is in a master device, and data associated with the peripheral identified by the chip select X is to be transmitted. In the case of a queued SPI of the prior art, an external controller must provide respective configuration commands for any data to be transmitted. Yet, with the SPI interface of the present invention, only the data and a relative address ADDR which corresponds to a certain address X of the memory section TRANSMIT RAM need be provided.

[0026] In fact, the address ADDR not only specifies to the interface that the data is to be stored in the memory section TRANSMIT RAM at the address X, but it also informs the interface that the data is to be transmitted to the peripheral X. Moreover, the block POINTER RAM receives the address on the bus ADDR and derives therefrom the addresses of the memory MODE RAM where the appropriate interface configuration commands are stored. These commands are used for transmitting the data to the peripheral identified by the chip select X.

[0027] If the interface of the present invention is used in a slave device, then the address on the bus ADDR pertains to the memory section RECEIVE RAM. It will be associated with an address of the memory MODE RAM at which the appropriate configuration command is stored to be executed in the receiving mode.

[0028] The interface of the present invention allows transmission and reception operations to be carried out without requiring that an external controller send an appropriate configuration command to the interface with each transfer. In this way, while preserving the possibility of varying the number and type of peripherals connected to the interface according to the user's needs, the structure of the external controller may be significantly simplified.